# RS-G2L100 User Guide

# Content

# 1. Yocto on the RS-G2L100

RS-G2L100 currently supports the system：Yocto.

# 2. Installing Yocto

## 2.1 Hardware tools required:

&#9312;　USB to TTL



&#9313;　HOST PC

&#9314;　The RS-G2L100 Board

&#9315;　12V power supply

## 2.2 Software tools required:

&#9316;　Tera Term

## 2.3 Step 1. Download the Yocto images and Tools from the website

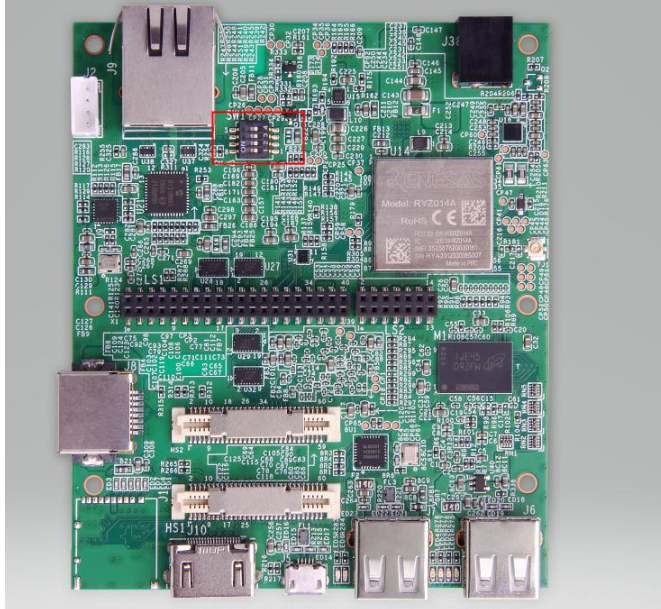1) Download the Yocto images from below website:

   https://mega.nz/file/CMwCkRoK#Ylf3IwSvw3KNjDOKR7en-ZMISkghy4wfRHxKOmYJBCg

2) Please download the Tera Term then install it in your computer from below website:

   https://www.filehorse.com/download-tera-term/
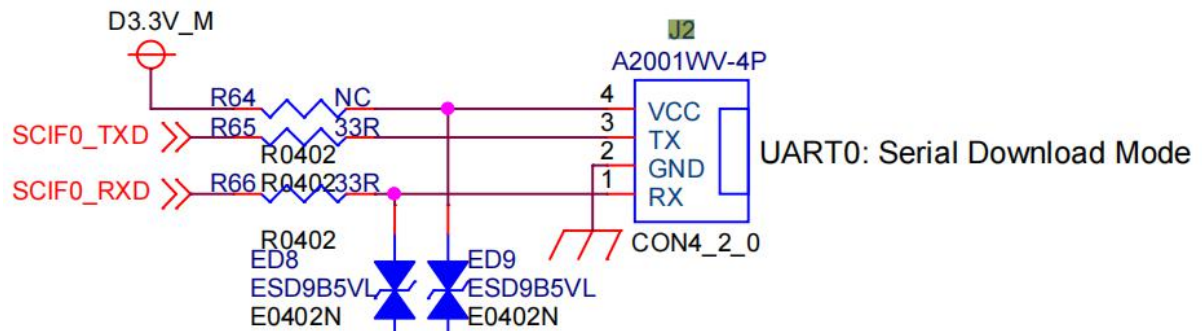
## 2.4 Step 2. Bring the board into burn mode

1) Set the start switch to: 1-off 2-on 3-off 4-off



2) USB to TTL :USB port is connected to the computer, and one end can be connected to the board through a DuPont cable, the serial port interface of the board is defined as follows:

## 2.5 Step 3. Flash the Yocto image

1) Connect the RS-G2L100 to the computer via USB to TTL.Open Tera Term, select Serial -> port: select the Port recognized by the computer to the USB to TTL;
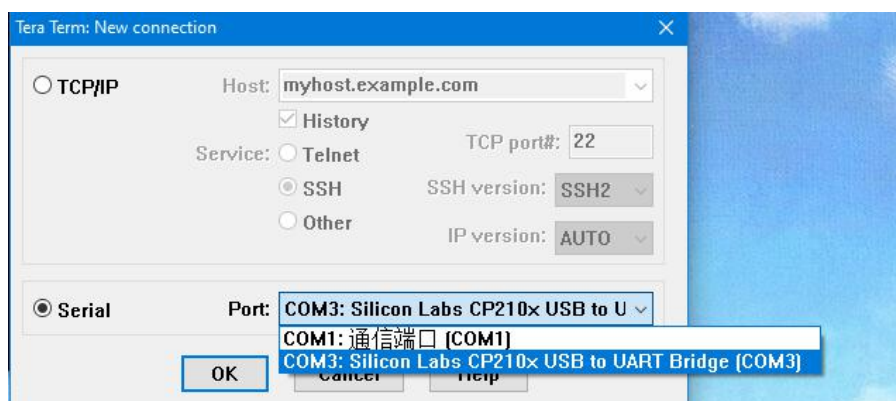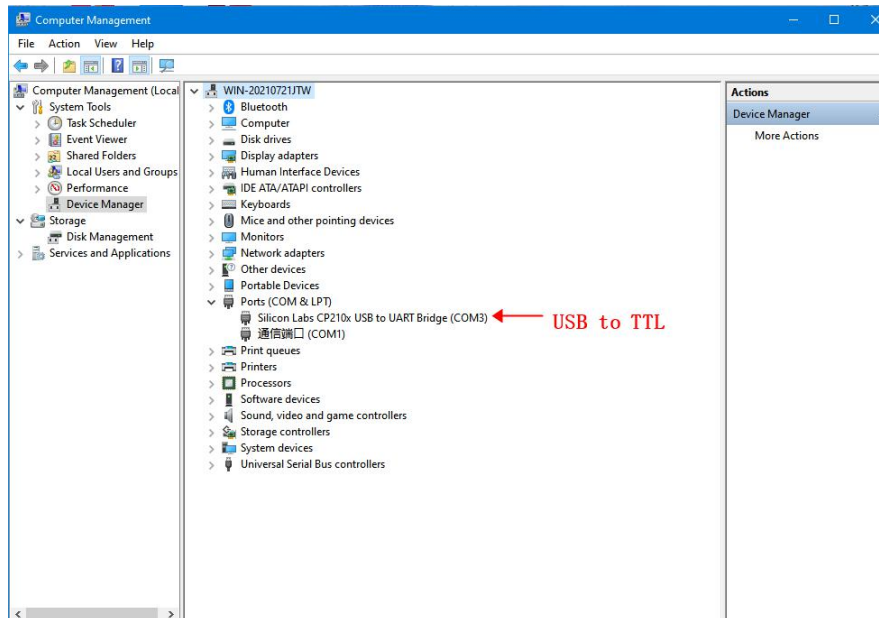




2) Unzip the Yocto images file.Set baud rate: Tera Term ->Setup->Serial port
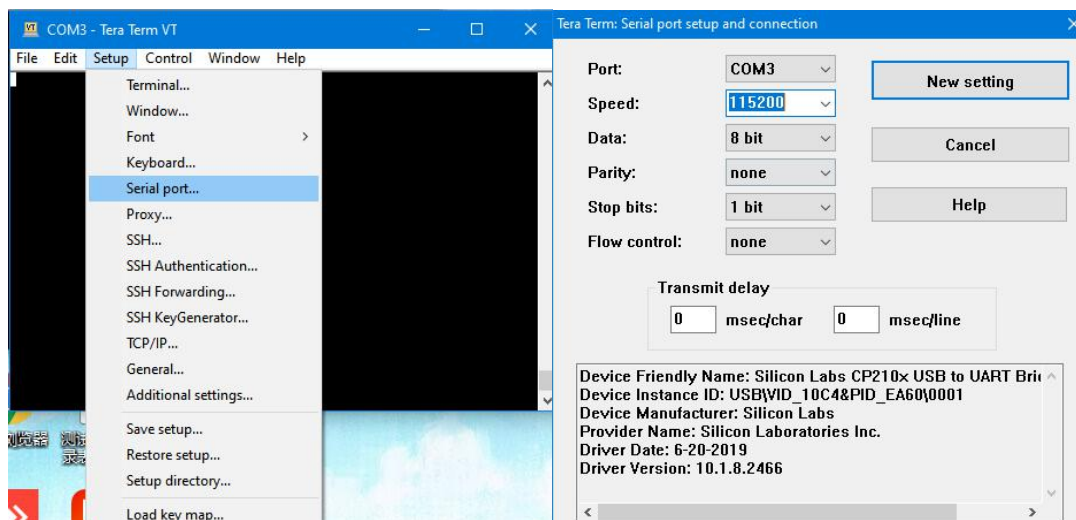
3) When everything is ready, plug in the device. Tera Term will display the

message:
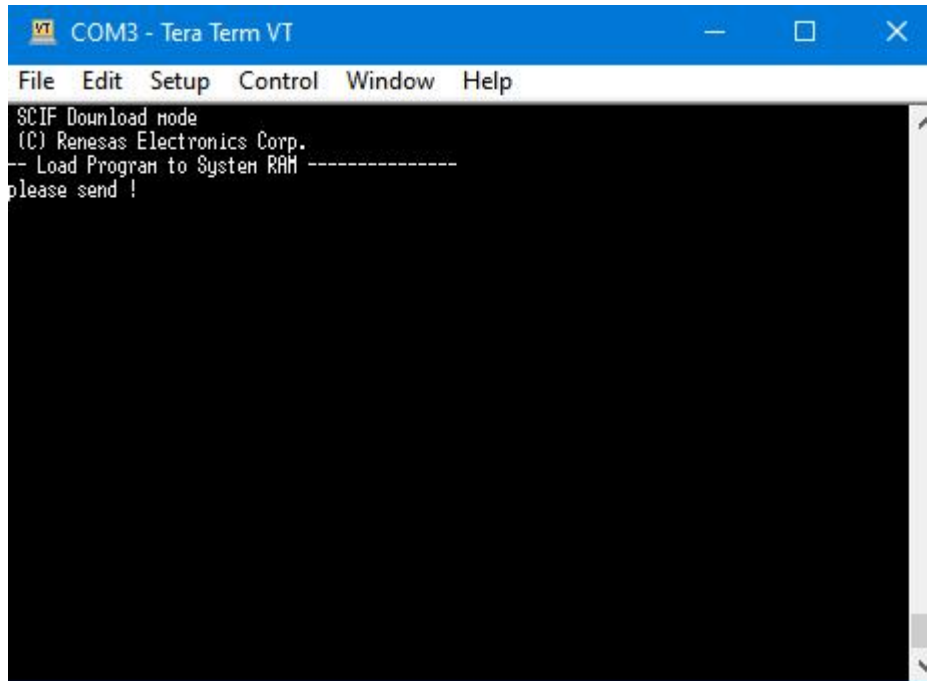


4) Send the

**Flash_Writer_SCIF_RZG2L_SMARC_PMIC_DDR4_2GB.mot** file to the

system.
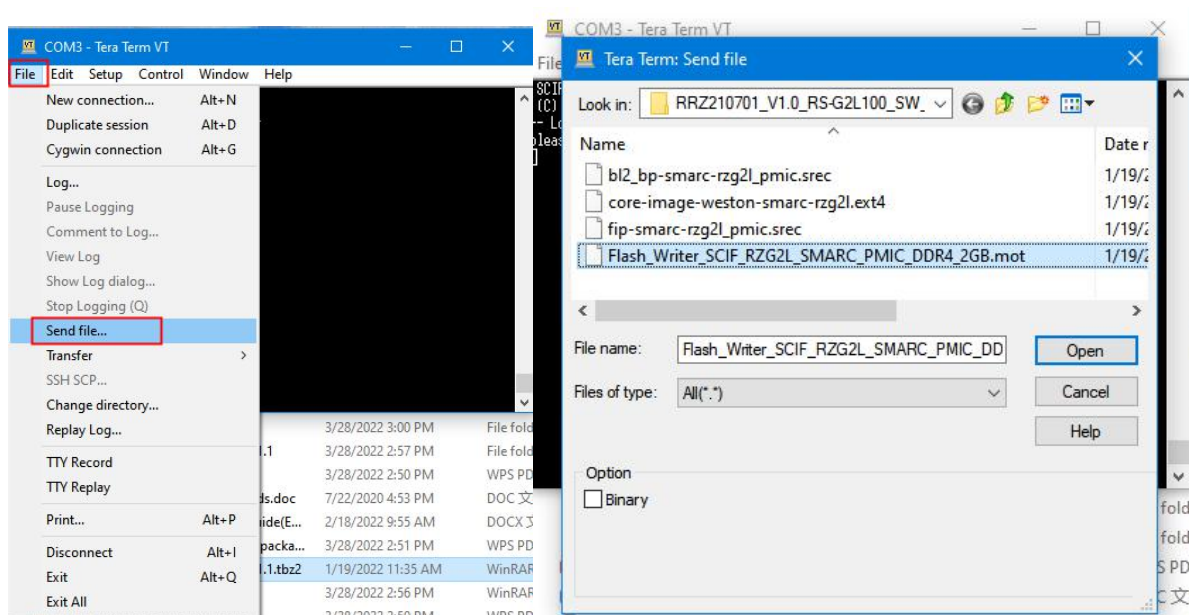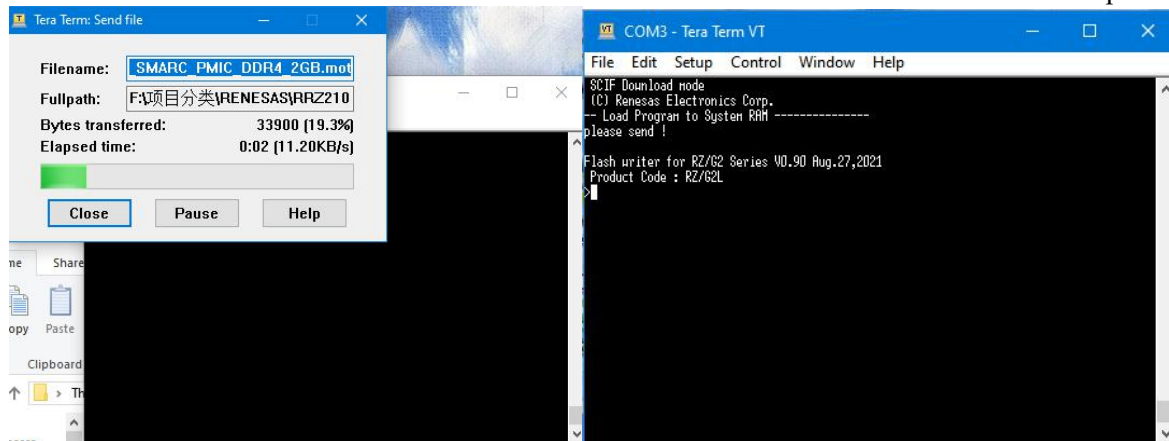
5) Write to file.(The red font is the input command)

>XLS2
===== Qspi writing of RZ/G2 Board Command ===============
Load Program to Spiflash
Writes to any of SPI address.
Micron : MT25QU512
Program Top Address & Qspi Save Address
===== Please Input Program Top Address =============
Please Input : H'11E00
===== Please Input Qspi Save Address ===
Please Input : H'0
Work RAM(H'50000000-H'53FFFFFF) Clear....
please send ! ('.' & CR stop load)



6) Send the **bl2_bp-smarc-rzg2l_pmic.srec** file to the system.
"SPI Data Clear(H'FF) Check : H'00000000-0000FFFF,Clear OK?(y/n)"
When you see the following prompt, please enter y and continue to execute.
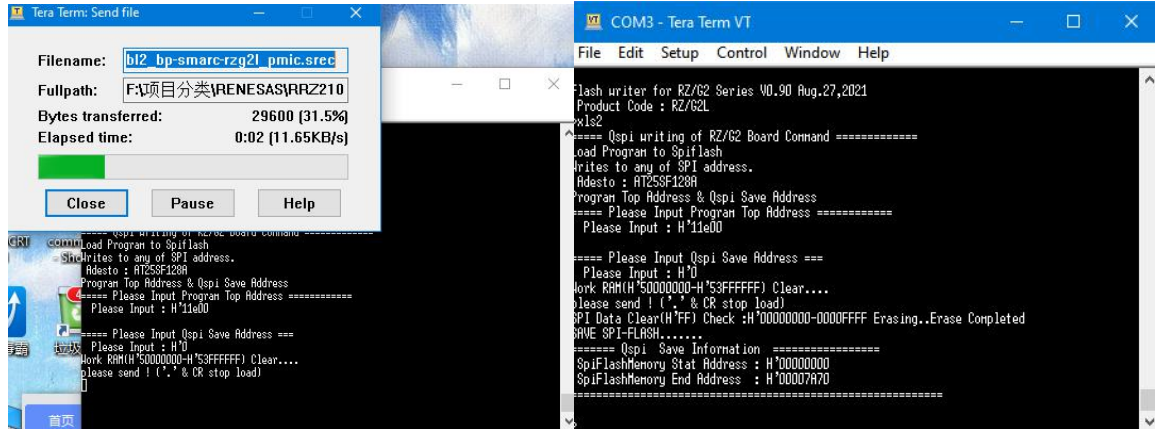
7) Write to file.(The red font is the input command)

>xls2

===== Qspi writing of RZ/G2 Board Command ==============

Load Program to Spiflash

Writes to any of SPI address.

 Adesto : AT25SF128A


Program Top Address & Qspi Save Address

===== Please Input Program Top Address =============

 Please Input : H'1d200


===== Please Input Qspi Save Address ===

 Please Input : H'0

Work RAM(H'50000000-H'53FFFFFF) Clear....

please send ! ('.' & CR stop load)

8) Send the **fip-smarc-rzg2l_pmic.srec** file to the system.Wait for the file upload to finish and the device is powered off.
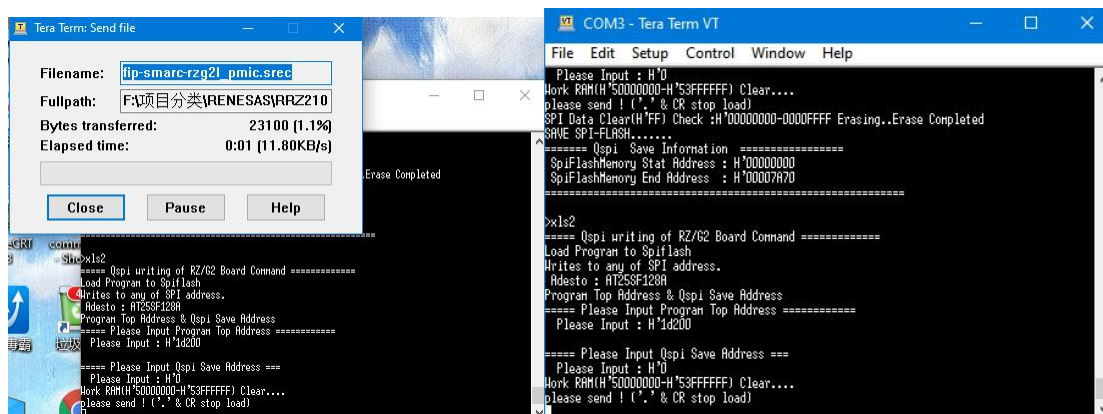


9) Put the board into boot mode: 1-off 2-off 3-on 4-off..Put the **core-image-**

**weston-smarc-rzg2l.ext4** file on the server.

10) RS-G2L100 to the computer via USB to TTL.Open Tera Term,after connecting to the power supply, keep pressing Enter on the keyboard to make the device enter uboot mode.

11) RS-G2L100 access network and set IP under uboot mode.(Please use this network port : J8)
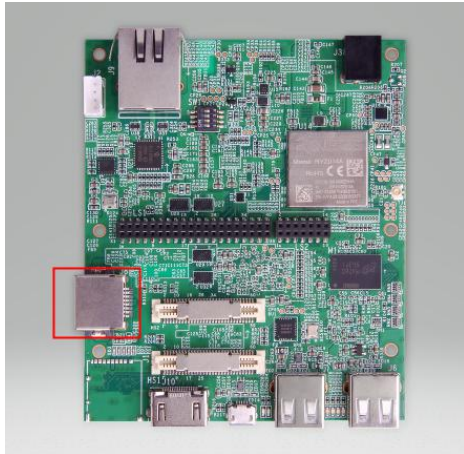


```
setenv ipaddr 10.168.4.104

setenv ethaddr 88:33:14:f6:c0:d4

setenv serverip 10.168.4.204

setenv gatewayip 10.168.4.1

saveenv
```

```
Model: smarc-rzg2l
DRAM:  1.9 GiB
MMC:   sh-sdhi: 0, sh-sdhi: 1
Loading Environment from MMC... OK
In:    serial@1004b800
Out:   serial@1004b800
Err:   serial@1004b800
Net:   eth0: ethernet@11c20000
Hit any key to stop autoboot:  0
=>
=>
=>
=>
=>
=>
=>
=>
=>
=>
=> setenv ipaddr 10.168.4.104
=> setenv ipaddr 10.168.4.104
=> setenv ethaddr 88:33:14:f6:c0:d4
=> setenv serverip 10.168.4.204
=> setenv gatewayip 10.168.4.1
=> saveenv
Saving Environment to MMC... Writing to MMC(0)... OK
```

12) Write to mmc .

```
tftpboot 0x58000000 core-image-weston-smarc-
rzg2l.ext4
mmc write 0x58000000 0 0x1317E4
```

```
############################################################
############################################################
############################################################
############################################################
############################################################
############################################################
############################################################
############################################################
############################################################
#########################
 2.2 MiB/s
done
Bytes transferred = 640665600 (262fc800 hex)
=> mmc write 0x58000000 0 0x1317E4

MMC write: dev # 0, block # 0, count 1251300 ... 1251300 blocks written: OK
=>
```

Note:0x1317E4 is the changed value. The value of the size of **ore-image-weston-smarc-rzg2l.ext4** needs to be divided by 512 and then converted to hexadecimal.

For example, the file size in the figure: 640665600 bytes /512=1251300  is converted to hexadecimal 1317e4.

## 2.6 step 4. Reboot and enjoy !

After completing the programming, power off and restart the device.(login:root)

```
Serial-COM3  x
[  OK  ] Started Load/Save Random Seed.
[  OK  ] Created slice system-weston.slice.
[  OK  ] Started Weston Wayland Compositor.
[  OK  ] Started Login Service.
[  OK  ] Created slice User Slice of UID 0.
[  OK  ] Reached target Multi-User System.
[  OK  ] Started Getty on tty2.
         Starting Update UTMP about System Runlevel Changes...
         Starting User Runtime Directory /run/user/0...
[  OK  ] Started User Runtime Directory /run/user/0.
[   6.450141] ravb 11c30000.ethernet eth1: failed to connect PHY
[  OK  ] Started Update UTMP about System Runlevel Changes.
[   6.486588] ravb 11c20000.ethernet eth0: failed to connect PHY
[  OK  ] Reached target Hardware activated USB gadget.
         Starting Save/Restore Sound Card State...
         Starting User Manager for UID 0...
[   6.604840] audit: type=1006 audit(1600598642.324:2): pid=211 uid=0 old-auid=4294967295
auid=0 tty=(none) old-ses=4294967295 ses=1 res=1
[  OK  ] Started Save/Restore Sound Card State.
[  OK  ] Reached target Sound Card.
[  OK  ] Started User Manager for UID 0.
[  OK  ] Started Session c1 of user root.

Poky (Yocto Project Reference Distro) 3.1.5 smarc-rzg2l ttySC0

BSP: RZG2L/RZG2L-SMARC-EVK/1.3
LSI: RZG2L
Version: 1.3
smarc-rzg2l login: [   33.848859] SDHI1 VccQ: disabling

smarc-rzg2l login:
smarc-rzg2l login:
smarc-rzg2l login:
smarc-rzg2l login:
smarc-rzg2l login:
```
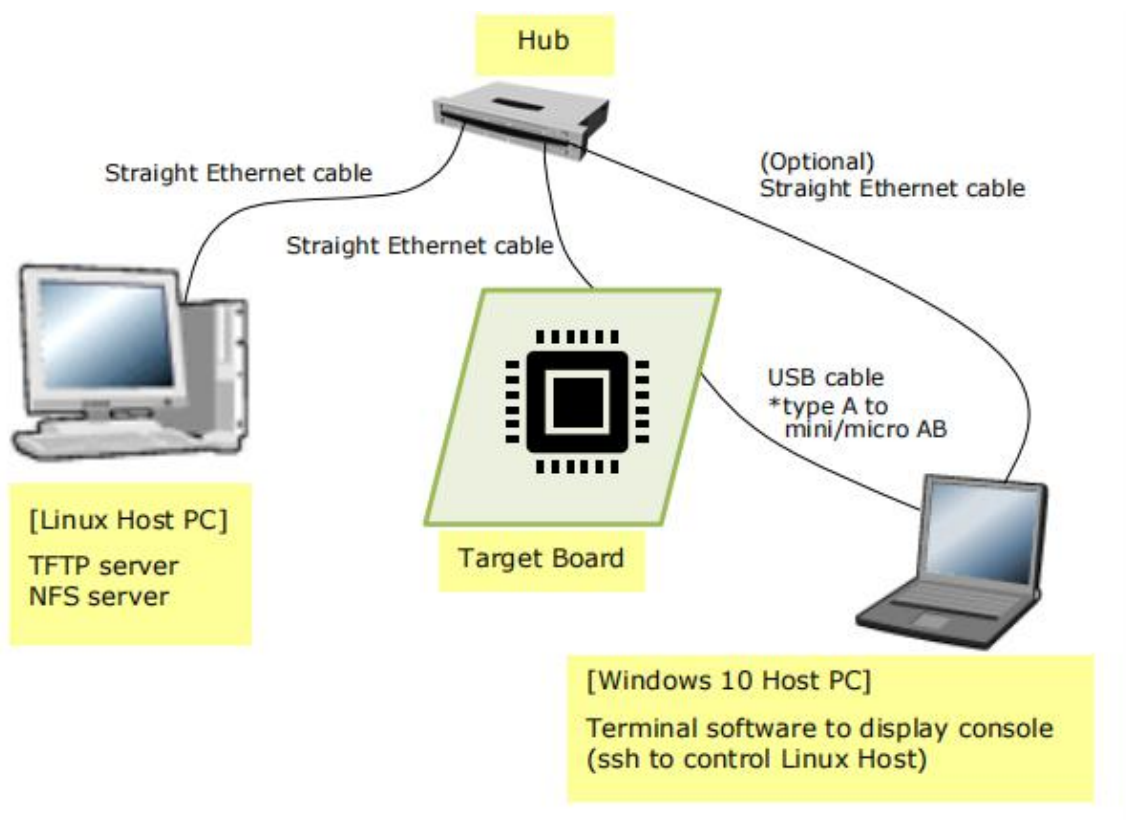
# 3. G2L00 Yocto Introduction

This section provides instructions for building and Yocto systems on the G2L100. If you need the source code, please contact us and we will publish the source code. Let's use our server as an example to compile.

## 3.1 Building Environment

This release note describes the contents, building procedures and important points of the G2L Board Support Package (hereinafter referred to as "BSP")

The recommended environment for BSP is shown below. The equipment and software used in this environment are listed in the following table.



| Equipment | | Description |
|---|---|---|
| Linux Host PC | | Used as build/debug environment |
| | OS | Ubuntu 20.04 LTS (recommended)<br>64 bit OS must be used. |
| | NFS server | Used for mounting rootfs via NFS |
| Windows Host PC | | Used as debug environment, controlling with terminal software |
| | OS | Windows 10 |
| | Terminal software | Used for controling serial console of the target board<br>Tera Term (latest version) is recommended<br>Available at https://ttssh2.osdn.jp/index.html.en |
| | VCP Driver | Virtual COM Port driver which enables to communicate Windows Host PC and the target board via USB which is virtually used as serial port. Available at:<br>• http://www.ftdichip.com/Drivers/VCP.htm |

**3.2 Building Instructions of a BSP**

**3.2.1 Building images to run on the board**

This section describes the instructions to build the BSP. Before starting the build, run the command below on the
Linux Host PC to install packages used for building the BSP.

```
$ sudo apt-get update
$ sudo apt-get install gawk wget git-core diffstat unzip texinfo gcc-multilib \
build-essential chrpath socat cpio python3 python3-pip python3-pexpect xz-utils \
debianutils iputils-ping python3-git python3-jinja2 libegl1-mesa libsdl1.2-dev \
pylint3 xterm
```

Please refer to the URL below for detailed information:
• https://docs.yoctoproject.org/3.1.5/brief-yoctoprojectqs/brief-yoctoprojectqs.html

Run the commands below and set the user name and email address before starting the build procedure. **Without this setting, an error occurs when building procedure runs git command to apply patches.**

```
$ git config --global user.email "you@example.com"
$ git config --global user.name "Your Name"
```

1) **Create a working directory (user_work) at your home directory, and decompress Yocto recipe package**
   Run the commands below. Copy the compressed Yocto recipe package files
   (rzg2l_bsp_v1.0.tar.gz) into your home directory prior to this step.

```
$ mkdir ~/user_work
$ cd ~/user_work
$ tar zxvf ~/rzg2l_bsp_v1.0.tar.gz
```

Note) When you use the WS1 version of the G2L LSI, please refer to the Section 3.2.3  Notes - (1) and apply the patches before the step (4) in the section 3.2.1.

2)  **Setup a build environment**
Run the commands below. The environment to build is set by the source command.

```
$ cd ~/user_work
$ source poky/oe-init-build-env
```

**3) Prepare the default configuration files for the target board**
   Run the commands below.

```
$ cd ~/user_work/build
$ cp ../meta-rzg2/docs/template/conf/minimal/smarc-rzg2l/*.conf ./conf/
```

**4) Start a build**
Run the commands below to start a build. Please connect the Linux Host PC to the network when building. Building an image can take up to a few hours depending on the user's host system performance.

```
$ bitbake core-image-minimal
```

After the build is successfully completed, a similar output will be seen, and the command prompt will return.

```
NOTE: Tasks Summary: Attempted 7427 tasks of which 16 didn't need to be rerun
and al l succeeded.
```

**3.2.2 Building SDK**
To build Software Development Kit (SDK), run the commands below after the steps (1) – (4) of section **3.2.1** are finished.
The SDK allows you to build custom applications outside of the Yocto environment, even on a completely different PC. The results of the commands below are 'installer' that you will use to install the SDK on the same PC, or a completely different PC.
For building general applications:

```
$ cd ~/user_work/build
$ bitbake core-image-minimal -c populate_sdk
```

The resulting SDK installer will be located in **build/tmp/deploy/sdk/**

The SDK installer will have the extension .sh
To run the installer, you would execute the following command:

```
$ sudo sh poky-glibc-x86_64-core-image-minimal-aarch64-smarc-rzg2l-toolchain-
\ 3.1.5.sh
```

### 3.2.3 Notes

**Early version of G2L Evaluation Board Kit**
When you use the WS1 version of the G2L LSI, please run the commands below to apply the patches before the step (4) in the section 3.2.1. When you want to know the details of the LSI version, please contact Renesas.

```
$ cd ~/user_work/meta-rzg2
$ patch -p1 < ../extra/0001-linux-renesas-add-workaround-patch-for-gicv3.patch
$ patch -p1 < ../extra/0002-tf-a-add-rd-wr-64-bit-reg-workaround.patch
```